Poster: The Curious Case of NTP Monlist

Teemu Rytilahti
Horst Görtz Institute for IT-Security (HGI)
Ruhr-Universität Bochum
Email: teemu.rytilahti@rub.de

Thorsten Holz
Horst Görtz Institute for IT-Security (HGI)
Ruhr-Universität Bochum
Email: thorsten.holz@rub.de

***Abstract*—While not a new threat, reflective amplification DDoS attacks exploiting vulnerable network services are still very prevalent despite the ongoing efforts to get them fixed.**

In this paper, we demonstrate how the very same feature used for NTP-based attacks can be used to form a global picture of ongoing attacks on the Internet. To this end, we first scanned the Internet to find vulnerable NTP servers, and subsequently requested their client lists hourly for a week. Our initial results suggest that only a fraction of all vulnerable services are currently suitable for attacks as well as for attack tracking. Furthermore, we show that there are many known vulnerable hosts which have remained unused due to their small response sizes, and argue that they may be abused for future attacks.

I. Introduction

Although distributed denial-of-service (DDoS) attacks as well as reflection-based attacks have been known for long [1], a combined form of such attacks leveraging vulnerable, mostly UDP-based (stateless) protocols such as Chargen, SSDP, DNS, and NTP [2] have become a common occurrence besides botnets. These so-called *reflective DDoS* attacks leverage vulnerable hosts to *reflect* much larger response packets (*amplification*) towards the victim by the means of spoofing the source IP address. NTP-based amplification attacks became notorious after being used for attacks in the beginning of 2014 [3] and continue to be a threat; according to Akamai's Q3/2015 report [4], NTP-based reflection attacks were still prevalent with almost 13 % share among DDoS attack methods. The culprit for these NTP-based attacks is a protocol extension called *monlist*, to which vulnerable hosts respond with information about their previous clients.

In this paper, we show how to infer the prevalence of attacks by doing regular, Internet-wide scans for vulnerable hosts to gather their client lists and other information. Based on this data, we can analyze the scale of such attacks and also reveal the existence of NTP servers which could be harnessed for attack purposes with modest effort.

II. NTP & Monlist Request

Network Time Protocol (NTP) is widely deployed protocol for synchronizing clocks over the Internet. Besides time synchronization, the protocol has implementation-specific extensions to allow querying peers, version as well as previously encountered clients, among other things. The `monlist` feature responds with a list of previous clients including information when they were last seen and how often they have been connecting. One 8 byte request is enough to trigger responses revealing details about up to 600 previously accessed clients, divided in up to 100 response packets. These responses contain information such as IP addresses, when the client has been last time in contact with the server, how often it has been seen, and how many requests there have been in total. All this is accompanied with per-connection information such as which mode, protocol version and port was used, which in turn makes the attack traffic generated by them interesting for analysis, with the goal to be able to reason about attacks made by leveraging vulnerable servers. Previous work by Czyz *et al.* [5] offers a more in-depth analysis.

III. Approach

Next we will describe the method we used to collect and filter out the attacks from our sample set.

1) Initial scan: First we gather a list of hosts with vulnerable NTP servers by sending a `monlist` request from a predefined port to all routable IP addresses with a custom-build scanner [6]. We save the incoming responses into a PCAP file, and collect a list of vulnerable servers to be used as a base-set for subsequent scans.

2) Gathering attack targets: After we have a list of vulnerable servers, we query only those listed servers to avoid burdening unnecessary hosts for our experiment. Each separate scan is launched from incremented port number in order to assign the responses to their correct time slots.

3) Attack filtering: As we have information about how often and when the client has been in contact with the server, we use this to filter out scanners and regular clients, and define an attack when: 1) a client is seen during the last hour, 2) the interval between requests is less than 100 seconds, and 3) a client has made more than 100 requests.

After this processing, we have a list of attacks gathered from a set of servers. We also define a host to be tracking-capable if there are clients with "mode 7" (which usually indicates the use of `monlist`) in its client list, thus over-estimating the amount capable and under-estimating non-capable hosts.

IV. Evaluation

We now analyze our results and reason why there is still unused attack potential.

A. Vulnerable servers

During our initial scan, we encountered total of 47,900 responsive hosts, from which approximately 72% (34,495)

TABLE I
SERVERS AND THEIR CLIENTS[1]

	Servers_N	Servers_T	Clients_N	Clients_T
count	197.0	197.0	18740.0	15755.0
mean	12334.6	6899.0	22.8	63.6
min	10931.0	4139.0	1.0	1.0
25 %	11641.0	5370.0	1.6	2.7
50 %	12099.0	6604.0	4.0	3.1
75 %	12900.0	8047.0	7.3	6.6
95 %	14164.0	10370.2	38.7	523.6
max	14683.0	11191.0	600.0	600.0

were responsive during our following scans. The amount of vulnerable servers varied (per time-point) during the experiment between 15,070 to 25,874 hosts, average being 18,704 hosts. Table I shows the distribution of servers seen hourly (count denotes # of time-points and # of servers, respectively), as well as the length of their client lists. What is notable here is that on average only about half of the servers at a given point of time were capable of tracking attacks.

B. Unleashed potential

As can be seen from Table I, there were at least almost nineteen thousand vulnerable hosts not returning any information about the usage of `monlist`. According to our analysis, some old ntpd versions do not insert information about `monlist` requests to the list, making our approach unsuitable for seeing the attacks, but at the same time unsuitable for attacking for now. To confirm our suspicions, we made regular NTP time synchronization requests to some of the available hosts to detect whether we can fill-up their `monlist` tables by regular requests, and that seems to be the case. Therefore, a malicious actor could take advantage of these non-tracking capable servers for attack purposes.

C. Attacks

After doing filtering as described above, we were left with 41,846 target hosts and 86,267 attacks done by 6,519 hosts, with average of 153 (median 38) clients per host. 7,579 non-globally routable IP addresses were seen, denoting that the feature is not only used maliciously. Table II displays the distribution of servers, attack duration, and the upper bound for potential traffic considering maximum response sizes without IP and UDP overheads, and is very unlikely to be seen in public due to various reasons. The median lengths of attacks seem to be decreasing due to the increasing use of so-called “booter services” (see report by Akamai [4]), which seems to correlate with these initial results.

TABLE II
ATTACK DISTRIBUTION

	Servers	Duration	Responses [2]
mean	52.8	3.5 hrs	485.1 GiB
std	79.7	11.4 hrs	54.5 TiB
50 %	3.0	1.9 hrs	364.6 MiB
85 %	122.0	4.2 hrs	14.1 GiB
95 %	212.0	8.0 hrs	203.3 GiB
99 %	346.0	24.2 hrs	2.3 TiB

V. LIMITATIONS

As our initial gathering of vulnerable devices was done over a period of two days, we have most likely missed some vulnerable hosts as well as recorded some more than once due to IP churn. This could be verified by doing fingerprinting [6], but was left out in this analysis. In order to gain more accurate information, the servers should be queried more often. The churn rate remains unknown, though.

The amount of vulnerable services being honeypots and/or not capable for generating real attack traffic (e.g. [7]) is unknown and that raises concerns over suitability of using server-provided information in the described way.

VI. CONCLUSION

The information provided by the attacking servers can be used as a tool to gather information about ongoing NTP amplification attacks, but it has limited use due to different server implementations as well as ongoing and past efforts to patch and shut down vulnerable NTP servers. This has caused their amount to decline (as observed by Kührer et al. [5], [6]), causing attackers to concentrate their attacks to be based on fewer amount of servers like analyzed by Akamai [4].

Our initial results show that there are plenty of usable, seemingly unused servers waiting to be used. One reason for those unfavored servers could be the fact that they do not have big enough client lists, but they could be abused for larger attacks by filling their client lists with regular queries before launching an attack. We leave the analysis of such hosts for future work while acknowledging that the proposed method for tracking attacks will not work on them.

REFERENCES

[1] V. Paxson, “An analysis of using reflectors for distributed denial-of-service attacks,” *ACM SIGCOMM Computer Communication Review*, vol. 31, no. 3, pp. 38–47, 2001.

[2] C. Rossow, “Amplification Hell: Revisiting Network Protocols for DDoS Abuse,” in *Symposium on Network and Distributed System Security (NDSS)*, 2014.

[3] M. Prince, “Technical details behind a 400Gbps NTP amplification DDoS attack,” http://blog.cloudflare.com/technical-details-behind-a-400gbps-ntp-amplification-ddos-attack, 2014.

[4] Akamai, “State of the Internet report, Q3 2015,” Tech. Rep., 2015.

[5] J. Czyz, M. Kallitsis, M. Gharaibeh, C. Papadopoulos, M. Bailey, and M. Karir, “Taming the 800 Pound Gorilla: The Rise and Decline of NTP DDoS Attacks,” in *Internet Measurement Conference*, 2014.

[6] M. Kührer, T. Hupperich, C. Rossow, and T. Holz, “Exit from Hell? Reducing the Impact of Amplification DDoS Attacks,” in *USENIX Security Symposium*, 2014.

[7] L. Krämer, J. Krupp, D. Makita, T. Nishizoe, T. Koide, K. Yoshioka, and C. Rossow, “AmpPot: Monitoring and Defending Against Amplification DDoS Attacks,” in *Research in Attacks, Intrusions, and Defenses*, 2015.

[1] T denotes tracking-capable and N non-capable servers

[2] Theoretical maximum based on “count” and assuming full client list